

RUTAS SEGURAS PARA UNA MEDELLIN MAS LIBRE DE ACOSO

<https://github.com/PedroPabloR20/ST0245-003>

Pedro Pablo Rodas	Isis Catitza Amaya	Santiago Alberto Roza Silva	Mauricio Toro
Universidad Eafit	Universidad Eafit	Universidad Eafit	Universidad Eafit
Colombia	Colombia	Colombia	Colombia
pprodasa@eafit.edu.co	icamayaa@eafit.edu.co	sarozos@eafit.edu.co	mtorobe@eafit.edu.co

Andrea Serna
Universidad Eafit
Colombia
asernac1@eafit.edu.co

RESUMEN

El acoso sexual es una problemática que hoy en día millones de mujeres tienen que sufrir a la hora de salir a la calle, ya sea para ir a su trabajo, descansar fuera de su hogar visitar otras personas, etc. Además, cada vez es más difícil no perderte cuando buscas una ruta para llegar a algún lugar, entonces porque no crear un algoritmo que nos ayude a solucionar ambos de estos inconvenientes.

Palabras clave

Camino más corto restringido, acoso sexual callejero, identificación de rutas seguras, prevención del crimen.

1. INTRODUCCIÓN

El mundo actual tiene miles de problemas con los cuales muchas personas sufren diariamente, entre todos estos problemas existe uno que ha estado aumentando su popularidad en la boca de muchos en la actualidad, este es el acoso en las calles, este existe de todo tipo pero bien sabemos que las mujeres son la que día a día la pasan peor con esta problemática y tiene que sufrir el miedo que es salir a la calle con la inminente idea de que algo les puede pasar si alguien en la calle decide abusar de ellas, por eso se quiere proponer una solución que ayudara a que las mujeres bien no esquiven al 100% el acoso, pero si se minimice de la mayor manera posible y que se les dé más tranquilidad a la hora de movilizarse por Medellín.

1.1. Problema

El principal problema al que se le quiere dar solución es el de tratar de reducir el peligro o más bien, evitar en lo mayor posible el acoso sexual al movilizarse por la ciudad (En pocas palabras moverme de un lado al otro con el menor riesgo posible)

1.2 Solución

Para la solución de este problema se ha escogido utilizar Dijkstra ya que este algoritmo tiene la capacidad y facilidad de calcular rutas con valor de peso, siendo así en este problema que propones solucionar, las rutas ya sean más cortas o con menos acoso generan un peso dentro de la solución entonces por medio de Dijkstra que es un algoritmo para el cálculo de rutas sin sobrepasar un peso restringido, es perfecto para la problemática a solucionar

1.3 Estructura del artículo

A continuación, en la Sección 2, presentamos trabajos relacionados con el problema. Posteriormente, en la Sección 3, presentamos los conjuntos de datos y los métodos utilizados en esta investigación. En la Sección 4, presentamos el diseño del algoritmo. Después, en la Sección 5, presentamos los resultados. Finalmente, en la Sección 6, discutimos los resultados y proponemos algunas direcciones de trabajo futuro.

2. TRABAJOS RELACIONADOS

A continuación, explicamos cuatro trabajos relacionados con la búsqueda de caminos para prevenir el acoso sexual callejero y la delincuencia en general.

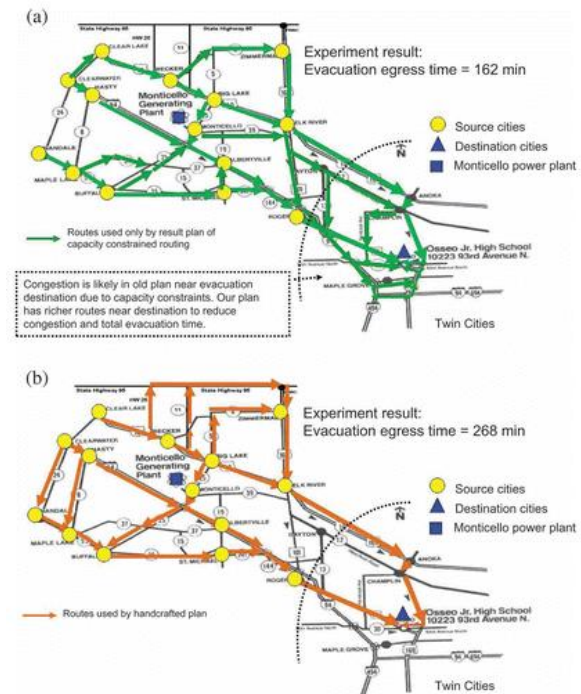
3.1 Ruta óptima para transporte de material peligroso.

Un grupo de programadores en Kuwait decidieron proponer una solución a una problemática de la cual había muchas vidas en riesgo, esta básicamente hablaba de como cuando se transportaba cierta mercancía que podía provocar catástrofes se podían encontrar en ciertas rutas obstáculos que dificultaban el transporte de estos materiales, por ende, proponen un algoritmo de análisis de rutas para identificar al momento de el transporte cual era la ruta más segura y con menos riesgo de accidente, como tal el algoritmo tomaba los datos de la topografía, clima, calles y lo que se encontraban en ellas, para así analizar cuál de las 4 posibles rutas que habían era la más segura y la que menos ponía en riesgo a la comunidad, el cual al concluir el experimento arrojó que las rutas 1 y 2 eran las más seguras ya que eran las que tenían menos cantidad de gente expuesta, más seguridad y en caso de emergencia más facilidad para evacuación, a diferencia de la 3 y 4 que tenían mismos índices de evacuación pero en cuanto a gente expuesta juntas sumaban un 83% de riesgo.



3.2 Experiencias del uso de algoritmos de análisis de rutas de evacuación

En este documento más que el hablarnos de una problemática específica con un problema de rutas, nos hablan de una solución previamente desarrollada y de los problemas que está a tenido a la hora de analizar las rutas para evacuar en caso de emergencia en la planta de energía, en el texto nos hablan de que aunque fue planeado de la forma más meticulosa posible, existen variables que no son posibles de analizar al momento ya que son variables que cambian cada segundo, dando que como se ve en la foto adjuntada aunque el experimento arrojo un tiempo de como seria la evacuación y el tiempo que esta llevaría si se usa el método que arroja el algoritmo, aun así, al experimentar con la realidad no es posible hacer esa evacuación de forma perfecta y por eso en el segundo caso arroja casi el doble de tiempo por ende se concluye al final del documento que aunque usar algoritmos para el trasado de rutas de evacuación es efectivo he útil se debe tener en cuenta como actuaran las personas en él y que no somos maquinas por lo que es mejor en la mayoría de casos usar rutas trazadas en gráficos, que las que son generadas dependiendo del momento.

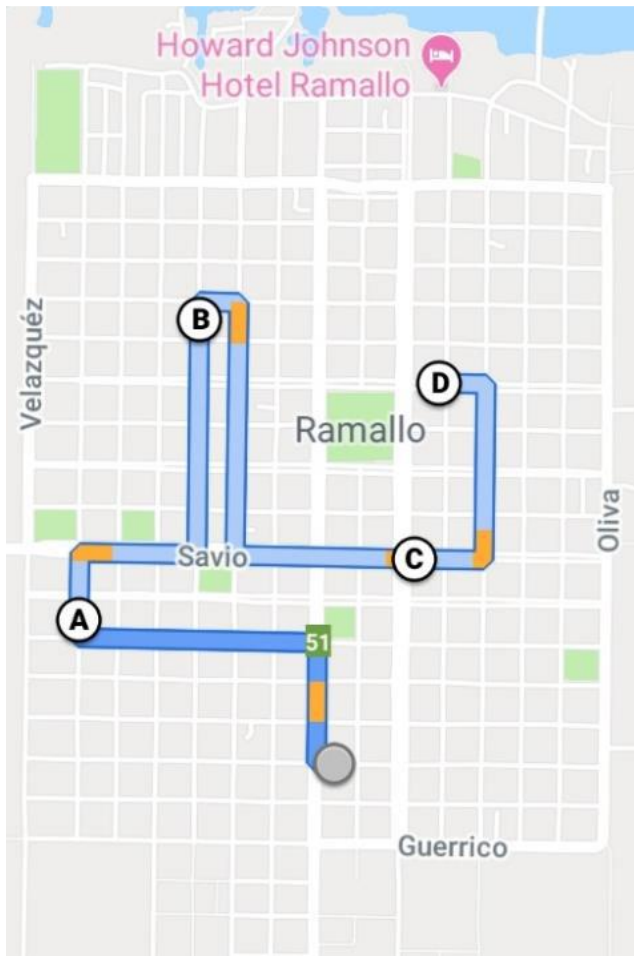


3.3 Algoritmo genético para la optimización de rutas de patrullaje

Siendo América Latina, una de las regiones más violentas del mundo, este caso se lleva a cabo en Argentina donde, hay tasas de hechos delictivos de 3391 casos por cada 100.000 habitantes, para contribuir a la disminución de delitos y la seguridad, se propuso utilizar una App, para los patrulleros que hace uso de Inteligencia Artificial y un Algoritmo Genético, esta muestra la ruta más óptima para el patrullaje, que se determina principalmente con los puntos más calientes en un área, y la distancia a vigilar.

Mas concretamente, el algoritmo funciona de la siguiente forma; se hace una generación de diez cromosomas lo cuales son un arreglo de puntos en la ciudad, combinado juntos con unos puntos calientes antes guardados, después a cada cromosoma se le asigna un puntaje, por último por medio del método elitista se escoge al cromosoma con mayor puntaje, y este se le enviara al patrullero a la App, que contendrá la ruta optimizada que patrullara; como dato adicional se harán veinte iteraciones de cromosomas escogiendo en cada iteración el mejor.

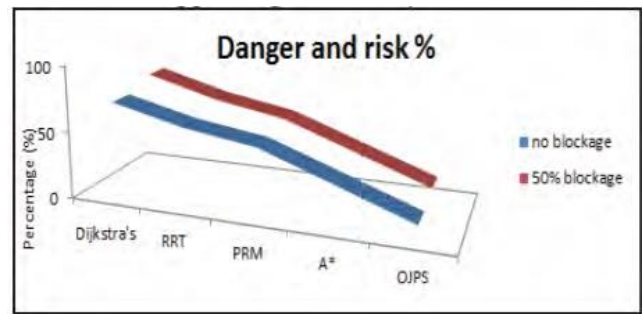
Los resultados se muestran a continuación en las siguientes imágenes.



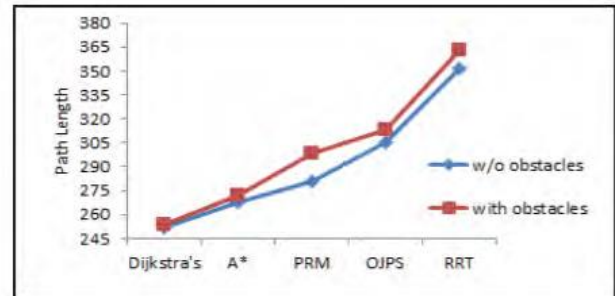
3.4 Rutas seguras para personas con discapacidad visual

Se parte del problema de accesibilidad que tienen las personas con alguna discapacidad visual, para circular a través de la ciudad. Las personas que padecen de alguna de estas dificultades visuales utilizan diversos elementos para transitar en la ciudad como bastones, pero este ofrece un rango de seguridad bajo, también existen otras alternativas como dispositivos para la detención de obstáculos, y entre otras soluciones más se encuentra los algoritmos para la planificación de la ruta mas segura para una persona con discapacidad visual.

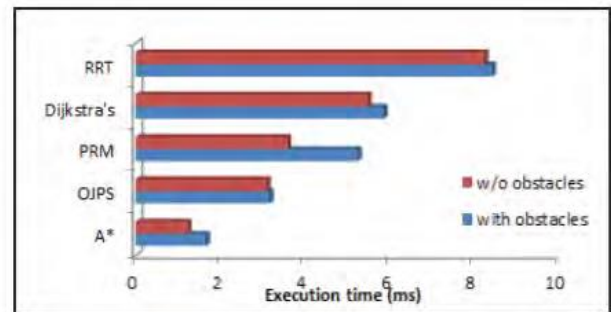
Existen una variedad de algoritmos para calcular una ruta con el problema antes descrito, entre ellos se encuentran A*, Dijkstra, Probabilistic roadmap (PR), Rapidly exploring Random Tree algorithm (RRT) y OrtogonalJump point search algorithm (OJPS). Para analizar cuál es mejor algoritmo se hicieron diferentes pruebas y comparaciones, se tomo la ruta mejor optimizada aquella, que hiciera el menor número de giros para la seguridad del discapacitado. Se muestran a continuación imágenes con los resultados de las pruebas.



Porcentaje de riesgo (giros) por cada algoritmo.



Longitud del camino por cada algoritmo (con y sin obstáculos)



Tiempo de ejecución para cada algoritmo.

Los resultados de las simulaciones muestran que el algoritmo OJPS es el que brinda una mayor seguridad en cuestión de giros realizados, Dijkstra es el que obtiene la ruta más corta y el algoritmo A* es el mejor en cuestión de tiempo de ejecución.

Como nota final, cabe aclarar que, aunque no se un caso similar enfocado en prevenir caminos con acoso sexual y delincuencia en general, si es parecido en buscar la mejor ruta con una restricción.

3. MATERIALES Y MÉTODOS

En esta sección, explicamos cómo se recogieron y procesaron los datos y, después, diferentes alternativas de algoritmos del camino más corto restringido para abordar el acoso sexual callejero.

3.1 Recogida y tratamiento de datos

El mapa de Medellín se obtuvo de Open Street Maps (OSM)¹ y se descargó utilizando la API² OSMnx de Python. La (i) longitud de cada segmento, en metros; (2) la indicación de si el segmento es de un solo sentido o no, y (3) las representaciones binarias conocidas de las geometrías se obtuvieron de los metadatos proporcionados por OSM.

Para este proyecto, se calculó la combinación lineal (CL) que captura la máxima varianza entre (i) la fracción de hogares que se sienten inseguros y (ii) la fracción de hogares con ingresos inferiores a un salario mínimo. Estos datos se obtuvieron de la encuesta de calidad de vida de Medellín, de 2017. La CL se normalizó, utilizando el máximo y el mínimo, para obtener valores entre 0 y 1. La CL se obtuvo mediante el análisis de componentes principales. El riesgo de acoso se define como uno menos la CL normalizada. La Figura 1 presenta el riesgo de acoso calculado. El mapa está disponible en GitHub³.

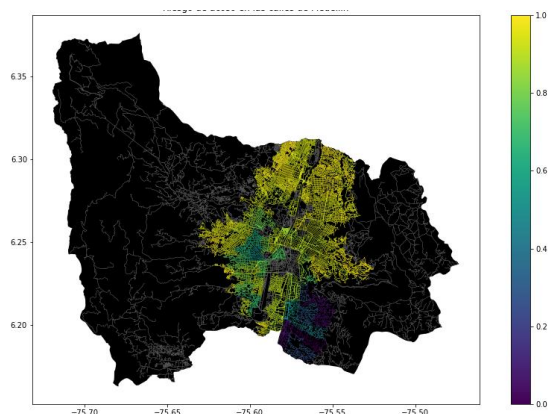


Figura 1. Riesgo de acoso sexual calculado como una combinación lineal de la fracción de hogares que se sienten inseguros y la fracción de hogares con ingresos inferiores a un salario mínimo, obtenida de la Encuesta de Calidad de Vida de Medellín, de 2017.

3.2 Alternativas de camino más corto con restricciones

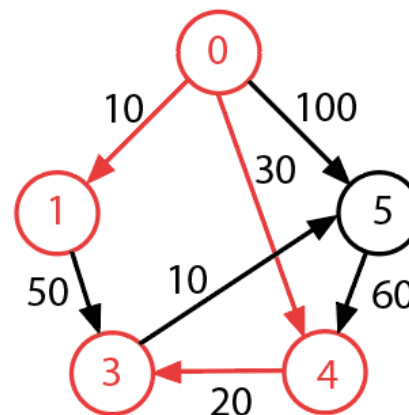
A continuación, presentamos diferentes algoritmos utilizados para el camino más corto restringido.

3.2.1 Dijkstra

El algoritmo de Dijkstra es el más sencillo y comúnmente utilizado a la hora de hallar el camino más corto en un

problema. El desarrollo que utiliza Dijkstra es simple, empezará desde un nodo raíz, pasando por los nodos óptimos, esto lo hará por medio de heurística, así ahorra tiempo al no tener que pasar por todos los nodos, hasta llegar al objetivo que es encontrar la ruta más corta.

La construcción que tiene el algoritmo es el siguiente; empezara con un nodo raíz, a continuación, se calcula la distancia entre el nodo raíz y lo vecinos no visitados, el vecino con menor distancia es a quien escogerá para pasar a los siguientes nodos, así seguirá hasta llegar al destino, con la ruta más corta.



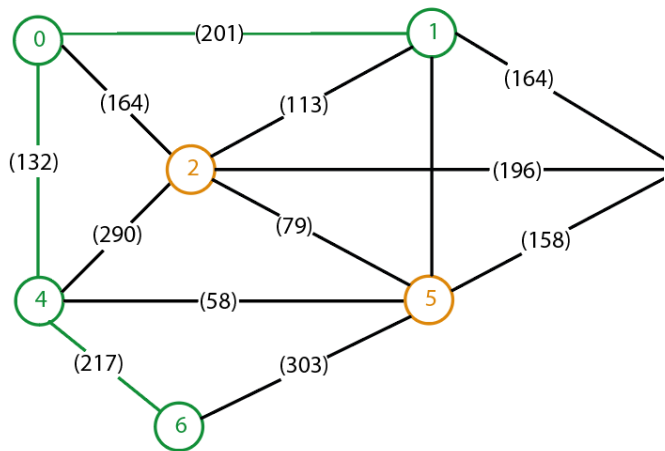
3.2.2 Árbol de peso mínimo

Este algoritmo da nacimiento a lo que conocemos como árbol de expansión de peso mínimo que comienza desde un vértice especificado dentro de un grafo y encuentra todos los vértices a los que tiene accesibilidad y determina el conjunto de relaciones que conectan los nodos con un valor de peso del menor tamaño posible.

¹ <https://www.openstreetmap.org/>

² <https://osmnx.readthedocs.io/>

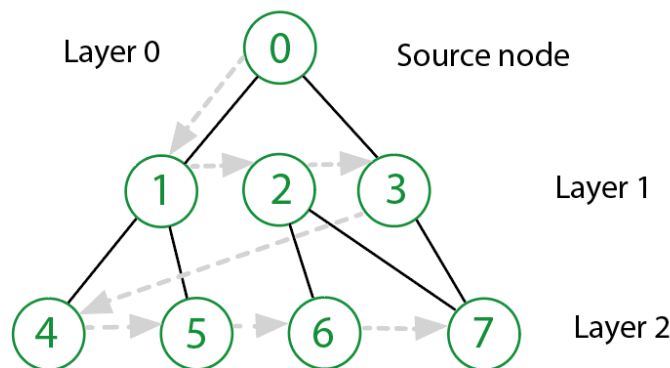
³<https://github.com/mauriciotoro/ST0245Eafit/tree/master/proyecto/Datasets/>



3.2.3 Breadth First Search

Este es uno de los dos algoritmos transversales de gráficos más fundamentales, su objetivo es encontrar el camino más corto. Por eso hoy en día es utilizado no solo para hallar recorridos normales, sino también en industrias como los GPS, los Motores de búsqueda, entre otros tipos de gráficos.

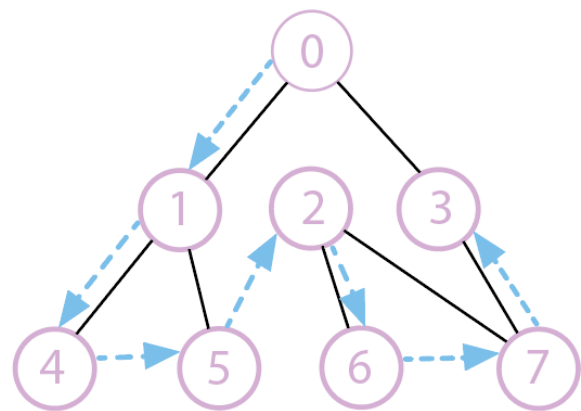
El algoritmo funciona de la siguiente manera; el algoritmo comienza su búsqueda desde la raíz (eligiendo algún nodo en caso de que sea un grafo) para explorar todos los vecinos de este nodo, luego se exploran todos los vecinos adyacentes y así hasta recorrer todo el grafo, no obstante, es importante resaltar que, si el algoritmo encuentra el nodo buscado antes de recorrer todos los nodos, termina su ejecución en ese instante.



3.2.4 Depth First Search

Este sería el otro algoritmo transversal de gráficos fundamental, igual de útil que el BFS anteriormente mencionado, este algoritmo se utiliza cuando queremos saber si cierta solución cumple con una cantidad específica de requisitos, como generar laberintos y recorrer arboles de forma jerárquica, no obstante, es importante notar que este algoritmo no necesariamente nos da como resultado la ruta más corta.

Su funcionamiento consiste en recorrer lo más lejos posible cada camino, antes de regresar, esto lo hace expandiendo cada nodo que va encontrado, de padre a hijo, así hasta llegar al final, luego, se devuelve y repite este procedimiento con cada uno de los vecinos adyacentes; al igual que en el algoritmo BFS, si se encuentra el nodo solicitado, antes de recorrer todos los nodos, el programa termina su ejecución en ese instante.



4. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO

A continuación, explicamos las estructuras de datos y los algoritmos utilizados en este trabajo. Las implementaciones de las estructuras de datos y los algoritmos están disponibles en Github⁴.

4.1 Estructuras de datos

La estructura de base en la que nos hemos basado para la realización de este proyecto es grafos esta tanto en matemáticas como en las ciencias de la computación es un conjunto de objetos a los cuales llamaremos vértices o nodos, los cuales están conectados por arcos o aristas, permitiendo tener relaciones entre los objetos de un conjunto. Desde un punto de vista práctico, los grafos permiten estudiar las interrelaciones entre unidades que interactúan unas con otras. Por ejemplo, una red de computadoras puede representarse y estudiarse mediante un grafo, en el cual los

⁴ [http://www.github.com/ ?????????? /.../proyecto/](http://www.github.com/?????????/.../proyecto/)

vértices representan terminales y las aristas representan conexiones (las cuales, a su vez, pueden ser cables o conexiones inalámbricas). Prácticamente cualquier problema puede representarse mediante un grafo, y su estudio trasciende a las diversas áreas de las ciencias exactas y las ciencias sociales.

La estructura de los datos se presenta en la Figura 2.

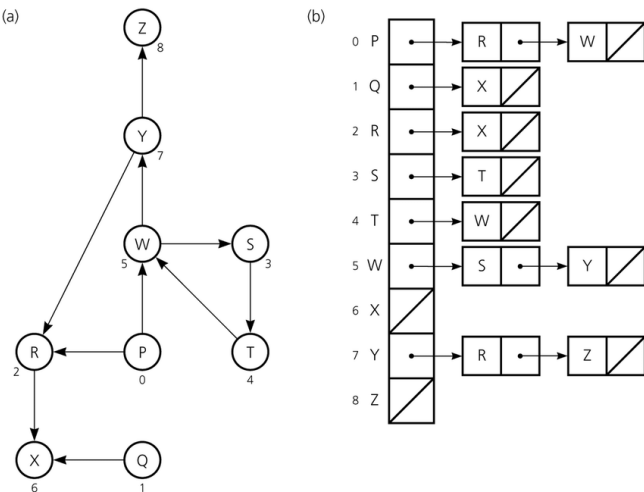


Figura 2: Un ejemplo de mapa de calles se presenta en (a) y su representación como lista de adyacencia en (b).

4.2 Algoritmos

En este trabajo, proponemos algoritmos para el problema del camino más corto restringido. El primer algoritmo calcula el camino más corto sin superar un riesgo medio ponderado de acoso r . El segundo algoritmo calcula el camino con el menor riesgo medio ponderado de acoso sin superar una distancia d .

4.2.1 Primer algoritmo

Para el primer algoritmo que el problema a solucionar era el camino más corto con sin superar una cantidad de riesgo establecida, utilizaremos el algoritmo de dijkstra, esta ya que como tal es el más útiles para calcular caminos cortos o caminos con restricciones, como tal este algoritmo funciona con valores de peso, donde cada arista de cada vértice tiene un valor, entonces el algoritmo como tal, se le da un valor que no debe de pasar y el traza la ruta con la cual se cumple el trayecto deseado y además no superando asi ninguna de las precondiciones, en el caso de nuestro proyecto es un camino más corto (d) que no supera un índice (r) de acoso

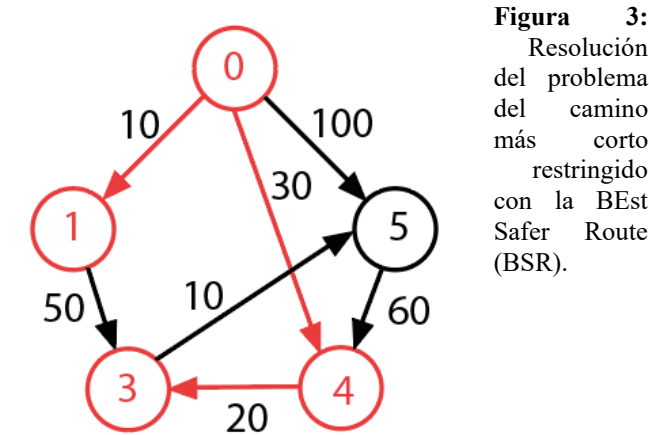


Figura 3: Resolución del problema del camino más corto restringido con la BEst Safer Route (BSR).

<https://github.com/PedroPabloR20/ST0245-003>

4.2.2 Segundo algoritmo

Explica el diseño del algoritmo para calcular el camino con el menor riesgo medio ponderado de acoso sin superar una distancia d y haz tu propia gráfica. No utilices gráficas de Internet, haz las tuyas propias. (En este semestre, el algoritmo podría ser DFS, BFS, una versión modificada de Dijkstra, una versión modificada de A^* , entre otros). El algoritmo se ejemplifica en la Figura 4.

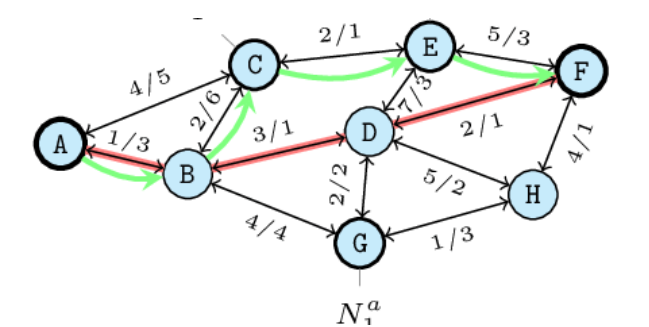


Figura 4: Resolución del problema del camino más corto restringido con la Búsqueda Primera Profunda (DFS). (Por favor, siéntase libre de cambiar esta gráfica si utiliza un algoritmo diferente).

4.4 Análisis de la complejidad de los algoritmos

Explica, con tus propias palabras, el análisis, para el peor caso, utilizando la notación O . ¿Cómo ha calculado esas complejidades? Explique brevemente.

Algoritmo	Complejidad temporal
Nombre del algoritmo	$O(V^2 * E^2)$
Nombre del segundo algoritmo (en caso de que haya probado dos)	$O(E^3 * V^2)$

Tabla 1: Complejidad temporal del nombre de su algoritmo, donde V es... E es... (Por favor, explique qué significan V y E en este problema).

Estructura de datos	Complejidad de la memoria
Nombre de la estructura de datos	$O(V * E * 2^E)$
Nombre de la segunda estructura de datos (en caso de que haya intentado dos)	$O(2^E * 2^V)$

Tabla 2: Complejidad de memoria del nombre de la estructura de datos que utiliza su algoritmo, donde V es... E es... (Por favor, explique qué significan V y E en este problema).

4.5 Criterios de diseño del algoritmo

Explique por qué el algoritmo fue diseñado de esa manera. Utilice criterios objetivos. Los criterios objetivos se basan en la eficiencia, que se mide en términos de tiempo y memoria. Ejemplos de criterios NO objetivos son: "estaba enfermo", "fue la primera estructura de datos que encontré en Internet", "lo hice el último día antes del plazo", "es más fácil", etc. Recuerda: Este es el 40% de la calificación del proyecto.

5. RESULTADOS

En esta sección, presentamos algunos resultados cuantitativos sobre el camino más corto y el camino con menor riesgo.

5.1.1 Resultados del camino más corto

A continuación, presentamos los resultados obtenidos para el camino más corto, sin superar un riesgo medio ponderado de acoso r , en la Tabla 3.

Origen	Destino	Distancia más corta	Sin exceder r
Universidad EAFIT	Universidad de Medellín	??	0.84
Universidad de Antioquia	Universidad Nacional	???	0.83
Universidad Nacional	Universidad Luis Amigó	??	0.85

Tabla 3. Distancias más cortas sin superar un riesgo de acoso medio ponderado r .

5.1.2 Resultados de menor riesgo de acoso

A continuación, presentamos los resultados obtenidos para el trayecto con menor riesgo de acoso medio ponderado, sin superar una distancia d , en la Tabla 4.

Origen	Destino	Acoso más bajo	Sin exceder d
Universidad EAFIT	Universidad de Medellín	??	5,000
Universidad de Antioquia	Universidad Nacional	???	7,000
Universidad Nacional	Universidad Luis Amigó	??	6,500

Tabla 3. Menor riesgo de acoso ponderado sin superar una distancia d (en metros).

5.2 Tiempos de ejecución del algoritmo

En la Tabla 4, explicamos la relación de los tiempos medios de ejecución de las consultas presentadas en la Tabla 3.

Calcule el tiempo de ejecución de las consultas presentadas en la Tabla 3. Indique los tiempos de ejecución medios.

	Tiempos medios de ejecución (s)
Universidad EAFIT a Universidad de Medellín	100.2 s
De la Universidad de Antioquia a la Universidad Nacional	800.1 s
De la Universidad Nacional a la Universidad Luis Amigó	845 s

Tabla 4: Tiempos de ejecución del nombre del algoritmo (Por favor, escriba el nombre del algoritmo, por ejemplo, DFS, BFS, un A* modificado) para las consultas presentadas en la Tabla 3.

6. CONCLUSIONES

Explique los resultados obtenidos. ¿Son los caminos más cortos significativamente diferentes de los caminos con

menor riesgo de acoso? ¿Qué utilidad tiene esto para la ciudad? ¿Son razonables los tiempos de ejecución para utilizar esta implementación en una situación real?

6.1 Trabajos futuros

Responda, ¿qué le gustaría mejorar en el futuro? ¿Cómo le gustaría mejorar su algoritmo y su aplicación? ¿Continuará este proyecto trabajando en la optimización? ¿En estadística? ¿Desarrollo web? ¿Aprendizaje automático? ¿Realidad virtual? ¿Cómo?

AGRADECIMIENTOS

Identifique el tipo de agradecimiento que desea escribir: para una persona o para una institución. Tenga en cuenta las siguientes pautas: 1. El nombre del profesor no se menciona porque es un autor. 2. No debe mencionar a los autores de los artículos con los que no se ha puesto en contacto. 3. Debe mencionar a los alumnos, profesores de otros cursos que le han ayudado.

A modo de ejemplo: Esta investigación ha sido apoyada/parcialmente apoyada por [Nombre de la Fundación, Donante].

Agradecemos la ayuda con [técnica particular, metodología] a [Nombre Apellido, cargo, nombre de la institución] por los comentarios que mejoraron en gran medida este manuscrito.

Los autores agradecen al profesor Juan Carlos Duque, de la Universidad EAFIT, por facilitar los datos de la Encuesta de Calidad de Vida de Medellín, de 2017, procesados en un *Shapefile*.

<https://github.com/PedroPabloR20/ST0245-003>

REFERENCIAS

- Fahad AlRukaibi, Duaij Alrukaibi, Sharaf Alkheder, Shahad Alojaiman, Tarek Sayed, Optimal route risk-based algorithm for hazardous material transport in Kuwait, Journal of Loss Prevention in the Process Industries, ISSN 0950-4230, <https://doi.org/10.1016/j.jlp.2018.01.012>.
- Shashi Shekhar and KwangSoo Yang and Venkata M.V. Gunturi and Lydia Manikonda and Dev Oliver and Xun Zhou and Betsy George and Sangho Kim and Jeffrey M.R. Wolff and Qingsong Lu, Experiences with evacuation route planning algorithms International Journal of Geographical Information Science, Volumen 26, Edición 12, Año 2012,

Taylor & Francis

<https://doi.org/10.1080/13658816.2012.719624>

- Eliseo S. Gamba, Valentina M. Domínguez, Alejandro M. Navarini “Algoritmos Genéticos: Optimización de rutas de patrullaje”. (2019) <https://rtyc.utn.edu.ar/index.php/ajea/article/view/528/425>
- Aragón, M. (2020). Revisión de la literatura de modelos de optimización para el tránsito de personas con discapacidad visual a través de rutas accesibles y seguras. Trabajo de investigación, Pontificia Universidad Católica del Perú, Lima. <http://hdl.handle.net/20.500.12404/20481>